

Article

Combining Color and Spatial Image Features for Unsupervised Image Segmentation with Mixture Modelling and Spectral Clustering

Branislav Panić , Marko Nagode * , Jernej Klemenc  and Simon Oman 

Faculty of Mechanical Engineering, University of Ljubljana, Aškerčeva Ulica 6, 1000 Ljubljana, Slovenia

* Correspondence: marko.nagode@fs.uni-lj.si; Tel.: +386-1-4771-507

Abstract: The demand for accurate and reliable unsupervised image segmentation methods is high. Regardless of whether we are faced with a problem for which we do not have a usable training dataset, or whether it is not possible to obtain one, we still need to be able to extract the desired information from images. In such cases, we are usually gently pushed towards the best possible clustering method, as it is often more robust than simple traditional image processing methods. We investigate the usefulness of combining two clustering methods for unsupervised image segmentation. We use the mixture models to extract the color and spatial image features based on the obtained output segments. Then we construct a similarity matrix (adjacency matrix) based on these features to perform spectral clustering. In between, we propose a label noise correction using Markov random fields. We investigate the usefulness of our method on many hand-crafted images of different objects with different shapes, colorization, and noise. Compared to other clustering methods, our proposal performs better, with 10% higher accuracy. Compared to state-of-the-art supervised image segmentation methods based on deep convolutional neural networks, our proposal proves to be competitive.

Keywords: spectral clustering; mixture models; color features; spatial features; image segmentation**MSC:** 68T10; 68T45; 62H30

Citation: Panić, B.; Nagode, M.; Klemenc, J.; Oman, S. Combining Color and Spatial Image Features for Unsupervised Image Segmentation with Mixture Modelling and Spectral Clustering. *Mathematics* **2023**, *11*, 4800. <https://doi.org/10.3390/math11234800>

Academic Editor: Konstantin Kozlov

Received: 24 October 2023

Revised: 19 November 2023

Accepted: 25 November 2023

Published: 28 November 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unsupervised image segmentation is a popular problem in many areas of mechanical engineering [1–4]. The examples range from the identification of porous structures [1] to the estimation of surface roughness and tool wear [2,3] or defect detection [5]. The goal is to extract the interesting pixels from the image, which are then further processed to obtain the desired information.

The usual image processing pipeline used in the above cases can be summarized in the following four steps [6]. The first step is image acquisition. The second step is image enhancement. Normally we need to manipulate the obtained raw image in some way, either by resizing, contrast enhancement, lighting correction, etc. In this way, the image is prepared for the image segmentation process, where segments of interest are found. These segments are finally used for the desired information extraction, e.g., surface roughness, tool wear, or defects. The third step is the image segmentation and the fourth step is the post-processing of the obtained segmentation. Post-processing can range from the simple calculation of the segment area to more sophisticated methods such as morphological operations to improve the resulting segmentation.

The choice of image segmentation method is usually determined by the presence of some kind of training dataset. Currently popular approaches to image segmentation, such as deep convolutional neural networks, require large training datasets but provide very

good results once the training dataset is available [2,5]. On the other hand, many conventional image processing methods such as gradient estimation and thresholding can be used as segmentation methods when no training dataset is available [1,3,6]. Conventional image processing methods are heavily dependent on pre- and post-processing steps, which usually need to be tailored to a specific problem and are therefore very difficult to automate. In the middle, which we call unsupervised image segmentation, are clustering methods, a form of machine learning that does not require a training dataset but is largely independent of pre- and post-processing steps, so they can usually be used without them [4].

A popular choice for unsupervised image segmentation is mixture modelling [7]. Mixture modelling is also interesting for unsupervised image segmentation because it can be used for further clustering [7,8] and also in conjunction with Markov random fields (MRFs), a powerful framework for image analysis [9]. Mixture models treat the image as a complex probability distribution consisting of a large number of simple parametric probability distributions that form a segment in the image. In this way, image segments are constructed that typically have similar pixel colors.

The problem with using mixture models for segmentation is that the segmentation is only based on the color information of the pixels and not on the spatial information. This can be mitigated by the subsequent use of MRF [10], but the segmentation is only improved locally and is generally suitable for noise reduction. Even with further clustering, such as the spectral clustering used in [7], the final segmentation is still only based on color pixel information. This can be satisfactory in some cases, but usually better results can be achieved if some kind of spatial constraints are included [11].

Therefore, in this work, we propose to combine the spatial image features and the color image features with spectral clustering to improve the image segmentation. Our main concept of the research methodology is shown in Figure 1. The spatial and color image features are obtained by mixture modelling. We introduce label moment estimation together with spatial and color adjacency matrix estimation. We also introduce some improvements in the estimation of the spatial MRF regularity matrix. Our proposals are tested on a self-generated image dataset, which also reflects the intended use of our proposals. Compared to the state of the art, our results are quite competitive.

On average, we achieved 10% higher accuracy than comparable clustering methods, which are mostly used for unsupervised image segmentation. Furthermore, for all generated images, we experimentally verified that using a different non-normal probability density function for the mixture model component has a negligible effect on performance. This is true even if the image was generated with a non-normal mixture model. It must be emphasized that this is true when the mixture model is to be used in the context of our proposal, i.e., when the components are to be merged using the proposed method. In other cases, the influence of a different probability density function is definitely not negligible [12]. Our results also show that using the MRF provides some relief for noisy images, but this should be further investigated. Finally, the comparison of our proposal with state-of-the-art deep convolutional neural networks used for supervised image segmentation shows that they perform better on average. However, considering that the influence of the training dataset is not negligible, our proposals offer an acceptable compromise, especially in cases where the obtained images are not affected by noise.

The present work is organized as follows. Section 2 is reserved for related work. Section 3 is reserved for image segmentation based on mixture models. Section 4 is reserved for the proposal of our method and the derivation of the estimates used. Section 5 explains the method for generating the image dataset and Section 6 contains the results and discussion. The last section, Section 7, summarizes our work with concluding remarks and provides some insights into future work possibilities.

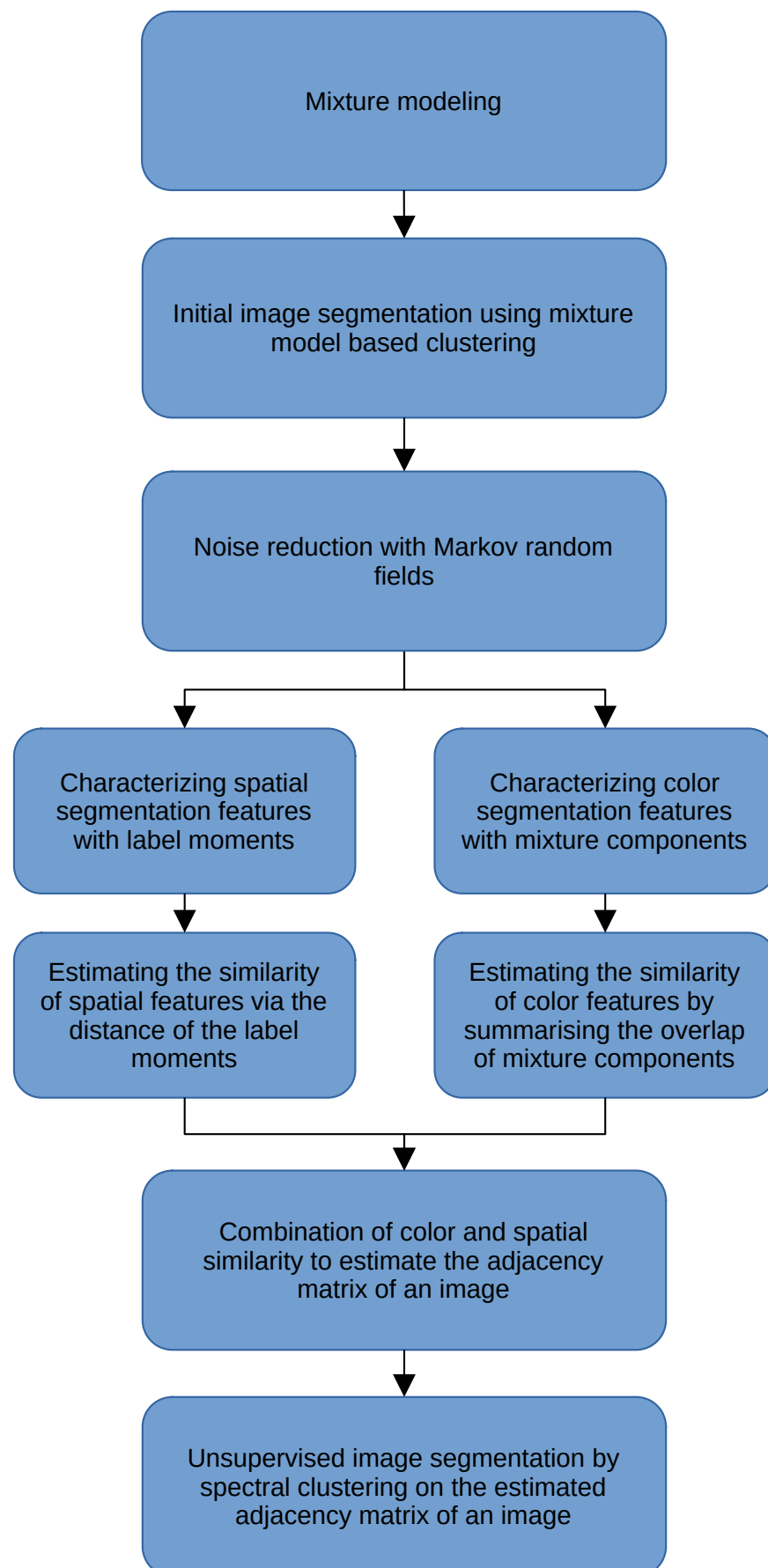


Figure 1. Main concept and methodology of our study presented in the form of a block diagram.

2. Related Work

Unsupervised image segmentation is usually performed with the help of clustering algorithms [13,14]. The most commonly used algorithms are k -means [15], fuzzy c -means [11], mean-shift [16], density-based clustering [17], spectral-based clustering [18,19], and mixture model-based clustering [20,21]. Recent studies have shown that mixture models actually provide a better way to segment an image [14], at least compared to fuzzy c -means and k -means clustering. On the other hand, methods such as density-based clustering, mean shift, and spectral clustering can offer some advantages and improvements in unsupervised image segmentation, but the computational efficiency of such algorithms is much higher than that of mixture models or k -means and fuzzy c -means clustering [13]. Spectral clustering is even more affected by this, since the construction of the adjacency matrix for an image with n pixels has a time complexity of $O(n^3)$ [22]. Moreover, recent studies show that the use of mixture models in combination with new superior supervised image segmentation methods such as deep neural networks is very powerful [23–26].

Using the mixture model for unsupervised image segmentation requires modelling the probability density function of the image, which is usually represented as an image histogram [8,20,21,27]. The image histogram is often complex and unique and its modelling is difficult. For this reason, various mixture models have been proposed. Recent studies include the non-extensive mixture models described in Ref. [14], the flexible regularized mixture model proposed in Ref. [20], the spatially restricted Student's- t mixture model in Ref. [21], the Laplace mixture model in Ref. [28], the Gumbel mixture model in Ref. [29], and so on. On the other hand, some authors treat the problem of the complex image histogram with a different strategy [7,30]. Since it is generally known (see, e.g., references [27,31]) that the mixture models are particularly good at approximating an arbitrary probability density function, especially as the number of components increases, the equality of the number of components and the number of segments in the image (or more generally the number of clusters in clustering) can be different [7,8,12,30]. The number of components in the mixture model is therefore estimated using information criteria [32,33], such as the Bayesian information criterion (BIC) [34], which is usually optimal for estimating any probability density function [27]. However, the number of components is then usually higher than the desired number of segments for the unsupervised image segmentation, and the problem is then translated into the mixture component merging problem [32]. The authors in Ref. [7] use spectral clustering for the subsequent merging of the components, while in Ref. [8], the authors propose hierarchical merging of mixture components for unsupervised image segmentation. The work in Refs. [30,35] is similar. Instead of merging the components of a mixture afterwards, each segment is modelled with a mixture model. If merging components can be described as a top-down approach for unsupervised image segmentation, then this approach can be described as a bottom-up approach. However, to summarize, all these approaches generally tend to relax the need for a complex-shaped probability distribution function as a mixture component by considering the image segment as multiple mixture components.

Finally, the use of spectral clustering in conjunction with mixture models proved to be a good method for merging components of a mixture model [7]. In general, spectral clustering proves to be a quite good method for unsupervised image segmentation [36]. Therefore, our main focus in this work is to fill the gaps in unsupervised image segmentation by using the mixture models and spectral clustering. The work in Ref. [7] relies only on color image features, while here we introduce both spatial and color-based features. Moreover, we continue the work presented in Ref. [8] by tackling the problem of hierarchical merging of mixture components with a more efficient algorithm. To model the spatial dependencies between different image segments, we have relied on the image moments. Recently, there has been some interesting work on using different image moments to improve image segmentation [37–39]. It has been shown that using different image moments, either via shape priors [37] or moment filters [38], can improve the results of supervised image segmentation methods, such as convolutional neural networks. Furthermore, we partially

address the Markov random fields [40], which seem to be a very important method for modelling spatial dependencies with mixture models for image segmentation. In this context, we propose a simple and effective way to circumvent the need for a spatial regularization parameter β by introducing the spatial regularization matrix B , which is fully estimated by measuring the overlap between the components of the mixture model.

3. Color-Based Image Segmentation Using Mixture Models

Let Z be the observed image with n pixels. The size of the image $n = \text{height} \times \text{width}$ defines the number of rows n_r and the number of columns n_c . Each pixel is therefore defined by the position x_j, y_j and the color value z_j (gray value). It should be noted that we focus here on monochromatic images, but an extension to RGB or other color image models is easily possible, with the only difference being the process of mixture estimation.

The first step is to estimate the optimal mixture model based only on the color intensity values. The digital image, which is represented as a matrix of size $n_r \times n_c$, is usually decomposed into a vector [8] of length n .

The mixture model is a composite probability distribution consisting of many simple parametric distributions such as normal, lognormal, gamma, etc. [41]. It is most easily described by its probability density function (PDF)

$$f(z_j|c, \mathbf{w}, \Theta) = \sum_{l=1}^c w_l f_l(z_j|\Theta_l), \quad (1)$$

where z_j is the observation, c the number of components, $\mathbf{w}$ the component weights, and Θ the component parameters. The weights w_l have the properties of the convex combination $w_l \geq 0 \wedge \sum_{l=1}^c w_l = 1$, which ensures that Equation (1) is a correct PDF.

The estimation procedure is described in detail in [27] and is mainly based on the standard model selection procedure with known information criteria and some improvements of the estimation algorithms presented in [42]. The estimation procedure provides the optimal number of components c in the mixture together with the component parameters and component weights. Using the estimated optimal mixture model and the Bayes assignment rule, which is commonly used to obtain a clustering solution with mixture models, the initial image segmentation is determined. However, the initial number of segments s may be less than the estimated number of components c due to the overlap between different mixture components, so that $s \leq c$.

Let L be the initial image segmentation. It has the same size $n_r \times n_c$ as the originally observed image Z , but the pixel color values are now replaced by the index of the component of origin. Using the Bayes allocation rule, for each pixel color intensity z_j , we find the index of component of origin l_j that maximizes the posterior probability τ_{jl} , i.e.,

$$l_j = \underset{\tilde{l}=1, \dots, c}{\operatorname{argmax}} \tau_{j\tilde{l}} \quad \forall j \in \{1, \dots, n\}, \quad (2)$$

where the τ_{jl} , the posterior probability, is the probability that the j th pixel arose from the l th component, estimated as

$$\tau_{jl} = \frac{w_l f_l(y_j|\Theta_l)}{\sum_{\tilde{l}=1}^c w_{\tilde{l}} f_{\tilde{l}}(y_j|\Theta_{\tilde{l}})}. \quad (3)$$

For example, the original image and the initial image segmentation for $s = c = 3$ can be represented as follows:

$$Z = \begin{bmatrix} 7 & 8 & 20 & 24 & 19 & 6 \\ 5 & 6 & 22 & 21 & 20 & 3 \\ 7 & 9 & 92 & 10 & 9 & 1 \\ 7 & 71 & 75 & 85 & 4 & 5 \\ 4 & 5 & 75 & 7 & 6 & 8 \end{bmatrix}, L = \begin{bmatrix} 1 & 1 & 3 & 3 & 3 & 1 \\ 1 & 1 & 3 & 3 & 3 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 & 1 \end{bmatrix}. \quad (4)$$

4. Spectral Clustering Based on Color and Spatial Features

Let A be the adjacency matrix between the pixels of the image. The adjacency matrix encodes pairwise similarities between the pixels and represents the image as a graph structure suitable for further graph clustering, which is the core of spectral clustering [22]. Since an image contains a large number of pixels, creating an adjacency matrix for all pixels is expensive and usually impractical [43]. Instead of using the original image to construct the adjacency matrix, we reduce the tedious problem of constructing the image graph to a mixture component merging problem, effectively reducing the $n \times n$ adjacency matrix of the image to a $c \times c$ adjacency matrix. In this way, we improve the time complexity of the spectral image clustering problem to $O(1)$ if we always choose the same model selection procedure for the optimal estimation of the mixture model.

There are at least two aspects when thinking about similarities in images, at least when we reason about the use of image segmentation in mechanical engineering. The first usually concerns color similarity. Image segments with similar color tend to belong to the same object. This is already partly taken into account in color-based image segmentation with mixture models. However, since color-based image segmentation based on mixture models is usually incomplete [7,8], it is generally best to apply additional clustering algorithms to the results obtained with mixture models. We also want to take into account some spatial similarities between the segments. For this purpose, we first need to define the spatial features of the segmentation and calculate the pairwise similarity between them.

4.1. Label Moment Estimation

Let $m^{p,q}$ be the raw image p, q -moment. The raw image moment can be estimated as follows [44]:

$$m^{p,q} = \sum_{j=1}^n x_j^p y_j^q z_j, \quad (5)$$

where x_j and y_j are the j th pixel coordinates, p and q are the moment orders, and z_j is the pixel gray intensity. We have decided to estimate different raw image moments up to order 1, i.e., four different combinations: $p = 0$ and $q = 0$, $p = 0$ and $q = 1$, $p = 1$ and $q = 0$, and $p = 1$ and $q = 1$. So four different moments.

To estimate the different raw moments for each segment in the obtained initial segmentation L , Equation (5) can be conveniently written as follows:

$$m_l^{p,q} = \sum_{j=1}^n x_j^p y_j^q I(l, l_j) \quad \forall l \in \{1, \dots, s\}, \quad (6)$$

where $I(l, l_j)$ returns 1 if the j th pixel belongs to the l th segment; otherwise, if $l \neq l_j$, the function returns 0. In this way, we obtain four different raw moments for each image segment, which we call raw label moments.

Then we estimate both x and y centroids for each label l :

$$\bar{m}_l^{1,0} = \frac{m_l^{1,0}}{m_l^{0,0}}, \quad \bar{m}_l^{0,1} = \frac{m_l^{0,1}}{m_l^{0,0}}. \quad (7)$$

Finally, for each label, we retain the values $\bar{m}_l^{1,0}$, $\bar{m}_l^{0,1}$, and $m_l^{1,1}$ and then standardize them to $\mathcal{N}(0, 1)$ to remove the effects of different moment scaling. The resulting label moment

vector $\mathbf{m}_l = (\bar{m}_l^{1,0}, \bar{m}_l^{0,1}, m_l^{1,1})^\top$. It should be emphasized that the vector elements represent Z-scores of the values specified in Equations (6) and (7).

4.2. Estimating Adjacency Matrix Based on Spatial Segment Features

Let A_s be the spatial adjacency matrix used for spectral clustering. The idea is that the matrix should capture pairwise spatial similarities between estimated segments in the initial image segmentation L . Therefore, the size is $s \times s$. The remainder s is the initial number of estimated segments.

The pairwise spatial similarity between two segments l and $\tilde{l}$ can be determined using the simple radial basis kernel function

$$K_{s,(l,\tilde{l})} = \exp\left(-\frac{\|\mathbf{m}_l - \mathbf{m}_{\tilde{l}}\|^2}{2\sigma^2}\right), \quad (8)$$

where $l = 1, \dots, s$, $\tilde{l} = l, \dots, s$, σ is a smoothing parameter that controls the exponential decay, and $\|\cdot\|^2$ is the Euclidean norm that transforms the vector into a real positive number. $K_{s,(l,\tilde{l})}$ thus provides the element $(l, \tilde{l})$ of the matrix A_s together with $(\tilde{l}, l)$, because $K_{s,(l,\tilde{l})} = K_{s,(\tilde{l},l)}$. Moreover, $K_{s,(l,l)} = 1$, which agrees well with our definition of the similarity problem, since each segment should be most similar only to itself. Thus, the final matrix A_s is symmetric positive definite.

4.3. Estimating Adjacency Matrix Based on Color Segment Features

Since the initial image segmentation L was created using a mixture model, we can also examine the similarity between the two mixture components used to label two different image segments. In other words, the color distribution of the image segment l is described by the distribution of the l -th mixture component, and the image segment $\tilde{l}$ is described by the distribution of the $\tilde{l}$ -th component. To determine the similarity of the color features between the two image segments, it is therefore sufficient to determine the statistical distance between the two mixture components [32].

In terms of mixture modelling and clustering, we usually summarize the clustering solution by measuring the overlap between different mixture components [45]. A larger overlap between mixture components usually occurs when more components are needed to estimate the probability density, but these are underrepresented in the clustering solution [8]. Therefore, measuring the overlap between different mixture components provides a good representation of the similarity between the mixture components and thus the similarity of the color features in the image.

Let $K_{c,(l,\tilde{l})}$ be the color similarity measure between the mixture component l and the mixture component $\tilde{l}$, which is responsible for the image segments l and $\tilde{l}$. Let A_c be the color adjacency matrix used for spectral clustering. Each matrix element $(l, \tilde{l})$ thus represents the color similarity between the segments $(l, \tilde{l})$ in the same way as we discussed spatial similarity, and can be estimated with the color similarity measure $K_{c,(l,\tilde{l})}$. Again identical to the spatial adjacency matrix A_s , the color adjacency matrix A_c must be symmetric and positive definite with all diagonal elements $A_{c,(l,l)} = 1$ and consequently with the trace $\text{tr}(A_c) = s$. According to the reports of [8], both entropy-based and misclassification-based similarity measures provide good results for the combination of mixture model components in unsupervised image segmentation. Nevertheless, the improvements proposed in [45] are also considered to ensure that the above conditions are met.

4.3.1. Entropy-Based Similarity Metric

Let $E_{l,\tilde{l}}$ be the resulting entropy of the combination of l and $\tilde{l}$ image segments. Using [33] we can estimate $E_{l,\tilde{l}}$ with

$$\hat{E}_{l,\tilde{l}} = -\sum_{j=1}^n \left(\tau_{lj} \log \tau_{lj} + \tau_{\tilde{l}j} \log \tau_{\tilde{l}j} \right) + \sum_{j=1}^n (\tau_{lj} + \tau_{\tilde{l}j}) \log(\tau_{lj} + \tau_{\tilde{l}j}), \quad (9)$$

where τ_{lj} denotes the component-specific posterior probabilities of Equation (3) for the color intensity of the j -th pixel. Then, based on Ref. [45], $E_{l,\tilde{l}}$ is scale-corrected based on the weights $w_l, w_{\tilde{l}}$ of the components l and $\tilde{l}$:

$$\hat{E}_{c,(l,\tilde{l})} = \frac{\hat{E}_{l,\tilde{l}}}{n(w_l + w_{\tilde{l}})}. \quad (10)$$

The final similarity metric (ent) is obtained by removing the weight bias, i.e.,

$$K_{c,(l,\tilde{l})} = \frac{\hat{E}_{c,(l,\tilde{l})}}{H_{l,\tilde{l}}}, \quad (11)$$

where $H_{l,\tilde{l}}$ is

$$H_{l,\tilde{l}} = \frac{w_l}{w_l + w_{\tilde{l}}} \log\left(\frac{w_l}{w_l + w_{\tilde{l}}}\right) + \frac{w_{\tilde{l}}}{w_l + w_{\tilde{l}}} \log\left(\frac{w_{\tilde{l}}}{w_l + w_{\tilde{l}}}\right). \quad (12)$$

4.3.2. Misclassification-Based Similarity Metric

Let $p_{l,\tilde{l}}$ be the misclassification probability between two mixture components. Based on [32,45], it can be estimated as

$$\hat{p}_{l,\tilde{l}} = \frac{\sum_{j=1}^n \tau_{lj} 1(\tau_{lj} > \tau_{\tilde{l}j})}{\sum_{j=1}^n \tau_{lj}}, \quad (13)$$

where $1(\tau_{lj} > \tau_{\tilde{l}j})$ is equal to 1 if $\tau_{lj} > \tau_{\tilde{l}j}$, and 0 otherwise. This metric is already correctly scaled and freed from weighting bias, and fulfills the condition that it results in 0 if the components do not overlap and 1 if the components overlap completely. However, this metric is not necessarily symmetric, which means that $p_{l,\tilde{l}} \neq p_{\tilde{l},l}$. To ensure that the metric becomes symmetric, we simply use a higher value. The similarity metric for the misclassification probability (demp) can thus be written as follows:

$$K_{c,(l,\tilde{l})} = K_{c,(\tilde{l},l)} = \max(\hat{p}_{l,\tilde{l}}, \hat{p}_{\tilde{l},l}). \quad (14)$$

4.4. Improving Initial Image Segmentation Using Markov Random Fields

Images affected by noise usually have a high number of distorted pixels. They are randomly distributed across the image and can severely affect the initial image segmentation with mixture models, as they introduce randomly scattered segments into the image. In addition, the initial image segmentation with mixture models may contain some segments that do not have a coherent spatial shape, are not visibly compatible with other segments, etc., even if no noise is visible (or present) in the image. Such segments are barely connected to other segments and provide little to no information gain or understanding required for the further steps.

To mitigate this problem somewhat, we use Markov random fields to improve the initial image segmentation. In this way, we enforce a certain spatial regularity for the obtained image segments [40]. A segmentation task is reformulated as the minimization of a suitable energy function

$$U(x_j | l_j) = u(x_j | l_j) + \beta \sum_{\tilde{j} \in N_j} v(l_j, l_{\tilde{j}}), \quad (15)$$

where $u(x_j | l_j)$ is a unary term that depends on the probability of assigning the j th pixel to the segment l_j , and $\beta \sum_{\tilde{j} \in N_j} v(l_j, l_{\tilde{j}})$ is a pairwise term that controls the spatial regularity. The image segmentation is based on the minimization of Equation (15), which can be

performed in many ways [40,46]. We have chosen the famous Iterated Conditional Modes (ICM) algorithm [47] because of its simplicity and straightforward parallelization.

The spatial regularity of the segment is determined (1) by the neighborhood of the j th pixel N_j , (2) by the potential function $v(l_j, l_{\tilde{j}})$, and (3) by the parameter β . For the potential function $v(l_j, l_{\tilde{j}})$, we choose the Potts model [40,46]:

$$v(l_j, l_{\tilde{j}}) = 1 - \delta(l_j, l_{\tilde{j}}), \quad (16)$$

where $\delta(l_j, l_{\tilde{j}}) = 1$ if and only if l_j is equal to $l_{\tilde{j}}$. For the neighborhood system, we choose a second-order neighborhood that has nine connections [40,46]. Finally, we replace the parameter β with the spatial regularity matrix B , which is based on the estimated color adjacency matrix A_c . The spatial regularity matrix B is estimated as

$$\hat{B} = \exp(-A_c). \quad (17)$$

In this way, we introduce a variable regularization for different pairs of segments found in the neighborhood of the j th pixel N_j based on their color similarity. The energy function is reformulated as

$$U(x_j | l_j) = u(x_j | l_j) + \sum_{\tilde{j} \in N_j} \beta_{(l_j, l_{\tilde{j}})} v(l_j, l_{\tilde{j}}), \quad (18)$$

where $\beta_{(l_j, l_{\tilde{j}})}$ is the element $(l_j, l_{\tilde{j}})$ of the matrix B , if $K_{c, (l_j, l_{\tilde{j}})} \rightarrow 0$ is the value of $\beta_{(l_j, l_{\tilde{j}})} \rightarrow 1$. This simply means that the penalty is smaller if the color similarity between the segments is greater, and that the penalty is greater if the color similarity between the segments is smaller.

4.5. Concatenating Color and Spatial Adjacency Matrix for Spectral Clustering

Let A be the final adjacency matrix used for spectral clustering and unsupervised image segmentation. We propose that the final adjacency matrix A is formed from both the spatial adjacency matrix A_s and the color adjacency matrix A_c . The idea behind this is that we can control the proportion of one of the two matrices in the construction of the final adjacency matrix A , so that the limiting cases where we only want unsupervised color-based image segmentation with spectral clustering $A = A_c$ and the opposite, where we only want unsupervised spatial image segmentation with spectral clustering $A = A_s$, are possible. Any other scenario in between should be possible.

Let α be the weighting parameter that controls the proportion of the color-based adjacency matrix A_c . The final adjacency matrix A can be written as

$$A = D^{1/2} P D^{-1/2}, \quad (19)$$

where P is the probability adjacency matrix with the element $(l, \tilde{l})$, defined as

$$P(l, \tilde{l}) = \frac{\alpha A_s(l, \tilde{l})}{\sum_{l=1}^s \sum_{\tilde{l}=l+1}^s A_s(l, \tilde{l})} + \frac{(1 - \alpha) A_c(l, \tilde{l})}{\sum_{l=1}^s \sum_{\tilde{l}=l+1}^s A_c(l, \tilde{l})} \quad (20)$$

and D is a diagonal degree matrix whose diagonal elements are defined as follows:

$$D(l, l) = \sum_{\tilde{l}=1}^s P(l, \tilde{l}). \quad (21)$$

Finally, we merge the initial s segments found on the image Z into a smaller and more informative final number of segments k by using the classical spectral clustering algorithm described in [22] and implemented in the package **rebmix** via the method `mergeLabels`. This step completes the unsupervised image segmentation. The method and the results

are shown in Figure 2. Pseudocode for the unsupervised image segmentation with our proposed method can be found in Algorithm 1.

Algorithm 1: Model selection procedure.

Input: Grayscale image Z , smoothing parameter σ , weighting parameter α , Boolean variable *useMrf* indicating whether MRF should be used for denoising, Boolean variable *useEntropy* indicating whether entropy-based similarity should be used for the color similarity metric, and the final desired number of segments in the image k ;

Output: Segmented image with k optimal segments;

```

1: Estimate the optimal mixture model using the model selection method [27];
2: Estimate the initial image segmentation  $L$  with the optimal mixture model [27];
3: if useMrf do:
4:     Optimize the MRF formulated with Equation (18) using the ICM algorithm [47];
5: end
6: Initialize the label moment matrix  $M$  of size  $s \times 3$ ;
7: foreach  $l \in \{1, \dots, s\}$  do:
8:     Estimate  $\bar{m}_l^{1,0}$ ,  $\bar{m}_l^{0,1}$  and  $m_l^{1,1}$  label moments using Equations (6) and (7);
9:     Store the estimated values of the  $\bar{m}_l^{1,0}$ ,  $\bar{m}_l^{0,1}$  and  $m_l^{1,1}$  label moments in the  $l$ -th row of the label moment matrix  $M$ ;
10: end
11: Normalize the columns of the matrix  $M$ ;
12: Initialize the spatial similarity matrix  $A_s$  of size  $s \times s$ ;
13: foreach  $l \in \{1, \dots, s\}$  do:
14:     Select the  $l$ -th row of the matrix  $M$  as  $\mathbf{m}_l$ ;
15:     foreach  $\tilde{l} \in \{1, \dots, s\}$  do:
16:         Select the  $\tilde{l}$ -th row of the matrix  $M$  as  $\mathbf{m}_{\tilde{l}}$ ;
17:         Estimate  $A_s(l, \tilde{l})$  using the Equation (8);
18:     end
19: end
20: Initialize the color similarity matrix  $A_c$  with the size  $s \times s$ ;
21: foreach  $l \in \{1, \dots, s\}$  do:
22:     foreach  $\tilde{l} \in \{1, \dots, s\}$ ;
23:         if useEntropy do:
24:             Estimate  $A_c(l, \tilde{l})$  using Equations (9)–(12);
25:         else do:
26:             Estimate  $A_c(l, \tilde{l})$  using the Equations (13) and (14);
27:         end
28:     end
29: end
30: Estimate the adjacency matrix  $A$  with Equations (19)–(21);
31: Finish the image segmentation using spectral clustering and the adjacency matrix  $A$ ;
32: end

```

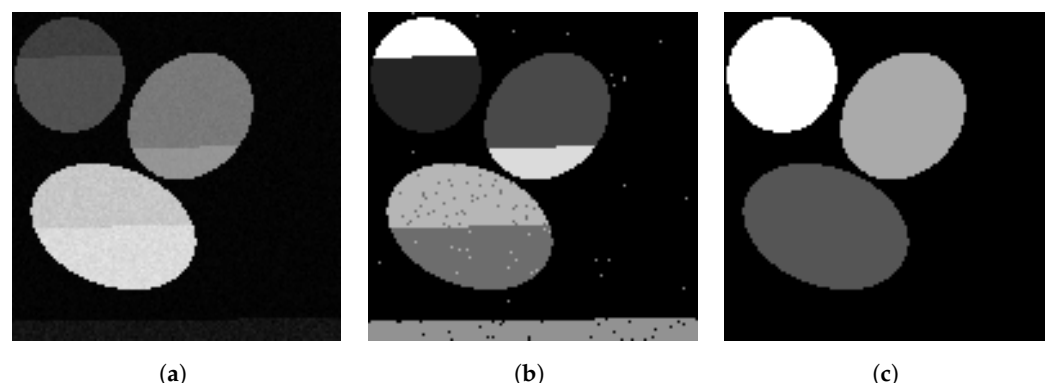


Figure 2. Illustration of results of proposed method for unsupervised image segmentation. (a) Original image. Image is filled with ellipses. Ellipses are populated with gray intensities from normal mixture model. (b) Initial image segmentation with normal mixture model. Number of initial segments is $s = 8$. (c) Final image segmentation using described procedure. Final number of segments $k = 4$.

5. Experimental Dataset

To illustrate the benefits of our proposals, we will use a simulated image dataset. The simulated image dataset aims at the task of segmenting images that contain a varying distribution of objects of different shapes and sizes. A similar image segmentation task is commonly used in many diagnostic problems in mechanical engineering [48] and model reconstruction [1].

The simulated dataset focuses on two-dimensional objects. The images are constructed as follows. First, the image is filled with objects of different shapes and sizes without overlapping (see Figure 3a). In this way, 50 images are created. These also represent the ground truth that we use to measure the success of our proposals. In the second step, we create artificial colorizations of pixels of different objects. We assign gray values from different mixture distributions in five different ways. The first colorization, shown in Figure 3b, assumes that the gray value of each pixel in the image comes from a normal mixture model. So each pixel gray value coming from each image object was taken from a normal mixture model with random number of components and random component parameters. In the same way, we apply the lognormal, gamma, and Weibull mixture model for the second, third, and fourth colorization (see Figure 3c–e). For the fifth colorization (Figure 3f), we also assume that the pixel gray intensities of an image object may belong to the normal mixture model, while some others may belong to the gamma mixture model. This makes the gray intensity distribution of the image more complex and difficult to approximate. The probability density functions of the different colorizations are shown in Figure 4b–f). Finally, we further increase the difficulty of image segmentation by applying Gaussian noise to the images of all scenarios (Figure 3g–k). In this way, we generate 500 different images for the test.

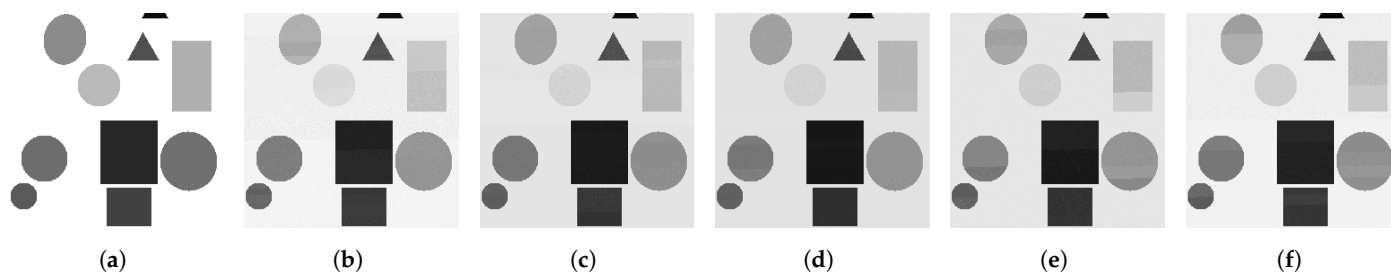


Figure 3. Cont.

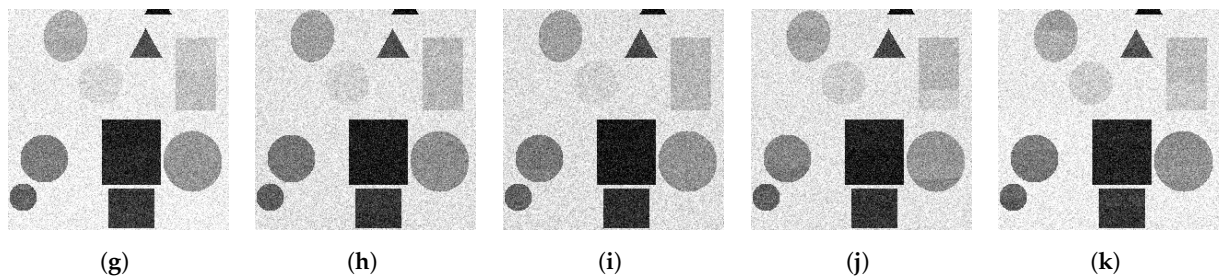


Figure 3. Original image along with different scenarios of colorizing pixel gray intensity values. (a) Original image. (b) First colorization. (c) Second colorization. (d) Third colorization. (e) Fourth colorization. (f) Fifth colorization. (g) First colorization with added noise. (h) Second colorization with added noise. (i) Third colorization with added noise. (j) Fourth colorization with added noise. (k) Fifth colorization with added noise.

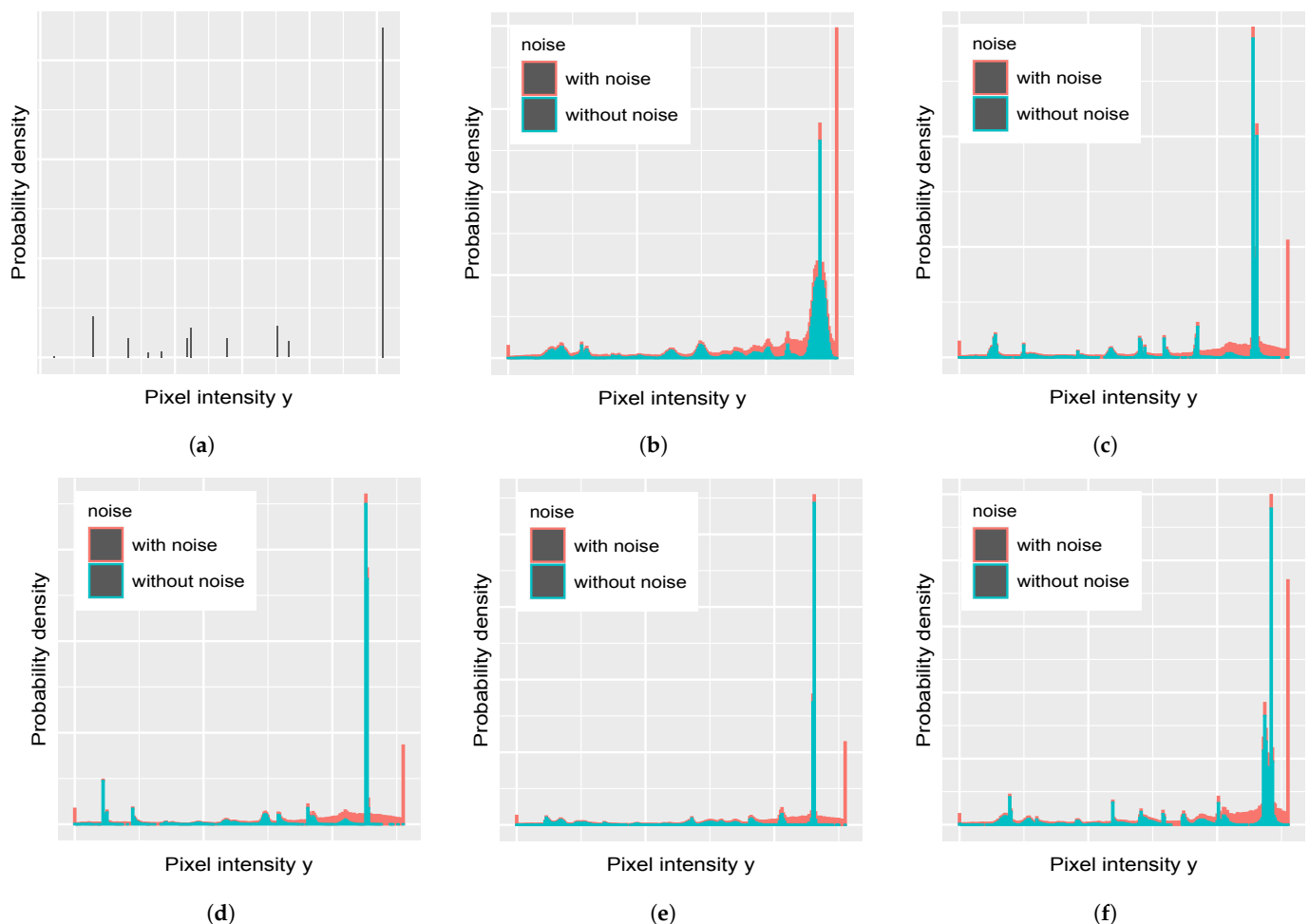


Figure 4. Pixel gray value distribution for the original image and different colorizations with and without noise addition. (a) Original image histogram. (b) Normal mixture colorization image histogram. (c) Lognormal mixture colorization image histogram. (d) Gamma mixture colorization image histogram. (e) Weibull mixture colorization image histogram. (f) Random mixture colorization image histogram.

6. Results and Discussions

For our experiment, we used the R package **rebmix** to estimate the optimal mixture parameters [49]. As in Panic et al. [27], the estimation strategy was single with the number of bins in the histogram $K = 255$. This is the best way to approximate the image histogram. Moreover, the maximum possible number of components in the mixture model was set

to $c_{\max} = 64$ and the information criterion for the optimal model selection was BIC. The ICM algorithm for Markov random fields was implemented in Python using the numba library [50], which allows the implementation of different algorithms for fast processing on GPU units. Finally, the methods `labelmoments` and `mergelabels` were added to the **rebmix** R package to further implement the label moment estimation and merging with spectral clustering described in this work.

6.1. Comparison of Different PDFs for Mixture Modelling with BIC Values

First, we focus on the results of the mixture modelling. Figure 5 shows the BIC values obtained for different image colorizations and different PDFs used for the parametric family distribution of the components of the mixture model, namely boxplots (distributions) of the BIC values with respect to the different colorizations, the PDF used for the mixture estimation, and the presence of noise. For example, the first colorization column yields eight different boxplots, four in the top panel and four in the bottom panel. The top panel shows four boxplots where the image used for the mixture estimation was not contaminated with noise. The bottom panel, on the other hand, shows four boxplots where the image used for mixture estimation was contaminated with noise. Finally, each boxplot in each panel was colored differently for the different component PDFs used for the mixture estimation.

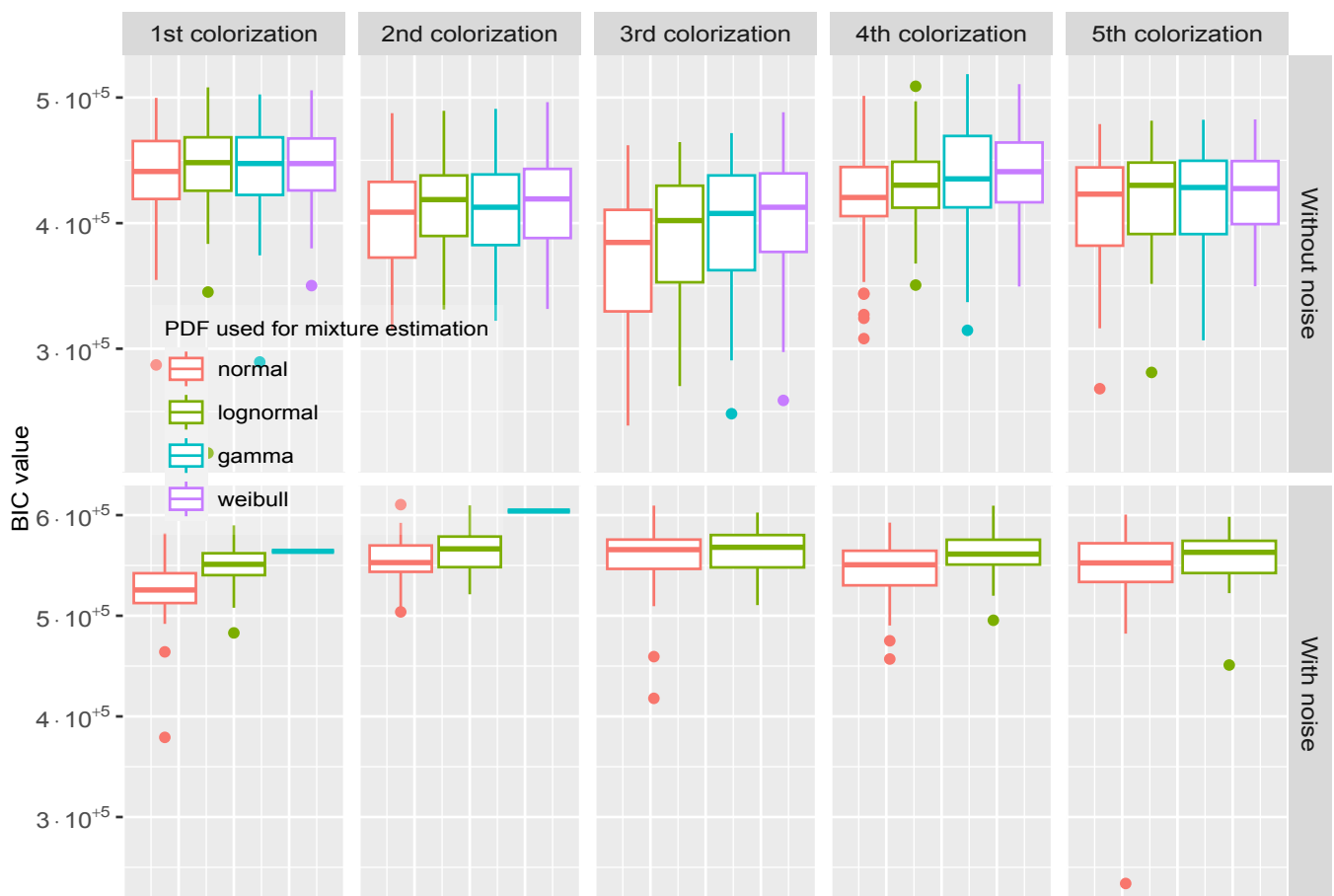


Figure 5. Boxplots of BIC values. The plots are grouped according to the PDF used for estimating the mixture model, image colorization, and the presence or absence of noise in the image.

Figure 5 shows that the normal mixture model is a very good approximator for different densities, as it usually has the best BIC values. For noisy images, the estimation procedure for the parametric gamma and Weibull families could not converge in most cases. This is due to the numerical procedure used for maximum likelihood estimation, which

does not have a closed form. Therefore, the available results in the presence of noise mainly refer to normal and lognormal PDF.

6.2. Comparison of Image Segmentation Accuracy between Different PDFs for Mixture Modelling and the Use of MRF

Our next reports deal with the accuracy of image segmentation. Here we use the adjusted rand index [51] for accuracy evaluation, which in this case corresponds to the classical accuracy definition for classification. We also report two figures, Figures 6 and 7. Figure 6 shows the boxplots of the accuracy values without using Markov random fields (MRFs) between the mixture modelling and the segment merging and with MRF as an intermediate step between the mixture modelling and the segment merging to improve the spatial coherence of the segmented image with mixture model.

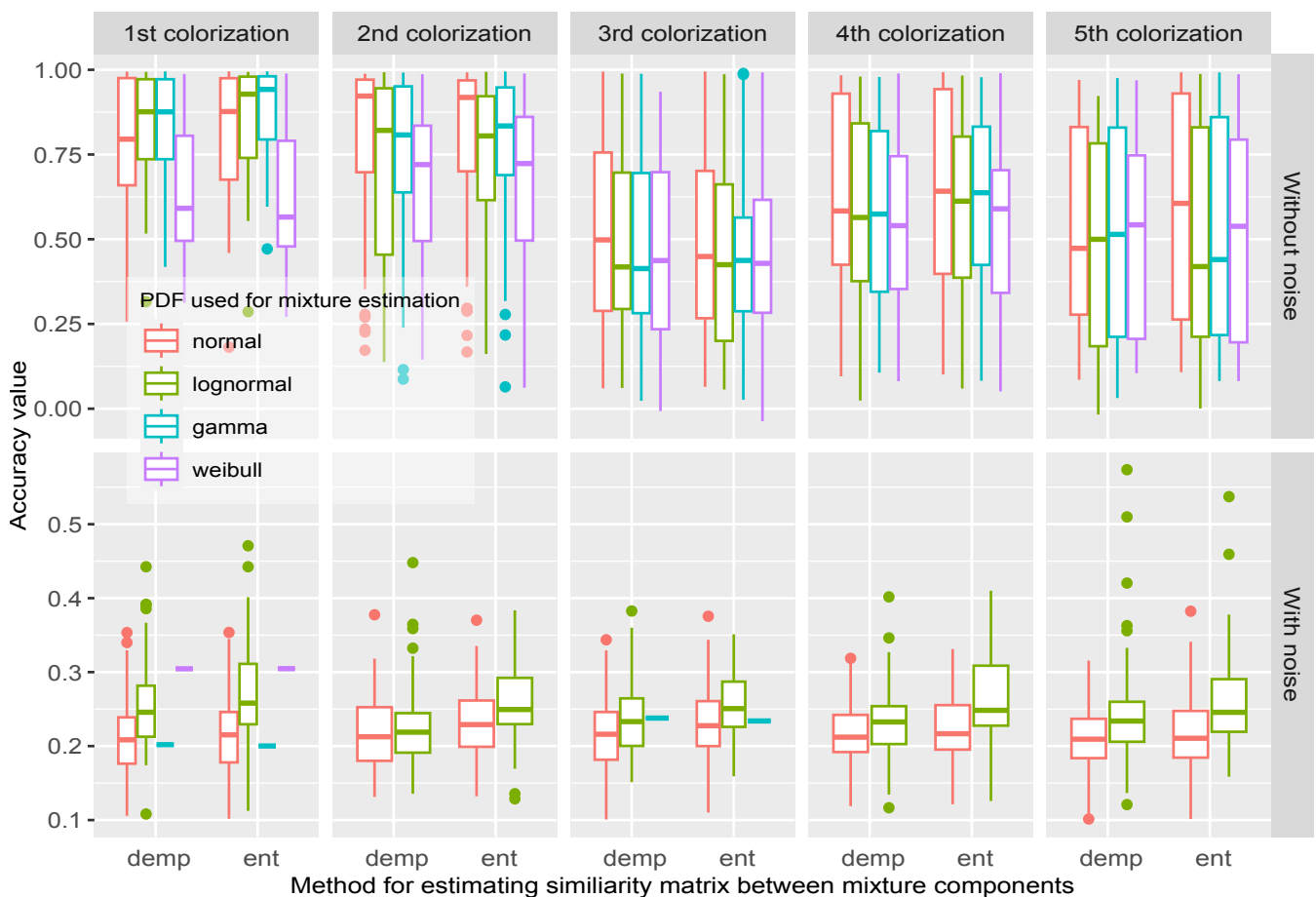


Figure 6. Boxplots of estimated accuracy values (ARI metric) without using Markov random fields (MRFs) as an intermediate step. Results are grouped by the PDF used to estimate the mixture model, the method used to estimate the similarity matrix (adjacency matrix), the absence/presence of noise on the image, and different colorization.

As can be seen in Figures 6 and 7, the presence of noise significantly degrades the results. In general, the estimated accuracy for a noisy image was twice as poor as for a noise-free image. It is also clear that using the MRF as an intermediate step between the mixture estimation and the segment merging improves the accuracy values very well. Regarding the PDF used for the mixture component, we can again conclude that the normal PDF is a rather strong competitor, which also agrees well with our results for the BIC values. Nevertheless, we can again conclude that the lognormal distribution is quite useful, especially for noisy images, where it shows the best accuracy values in most cases. Finally, the choice of estimation method for the similarity matrix of the mixture components does

not seem to play a significant role. This is especially true in the case where the Markov random fields are used as an intermediate step.

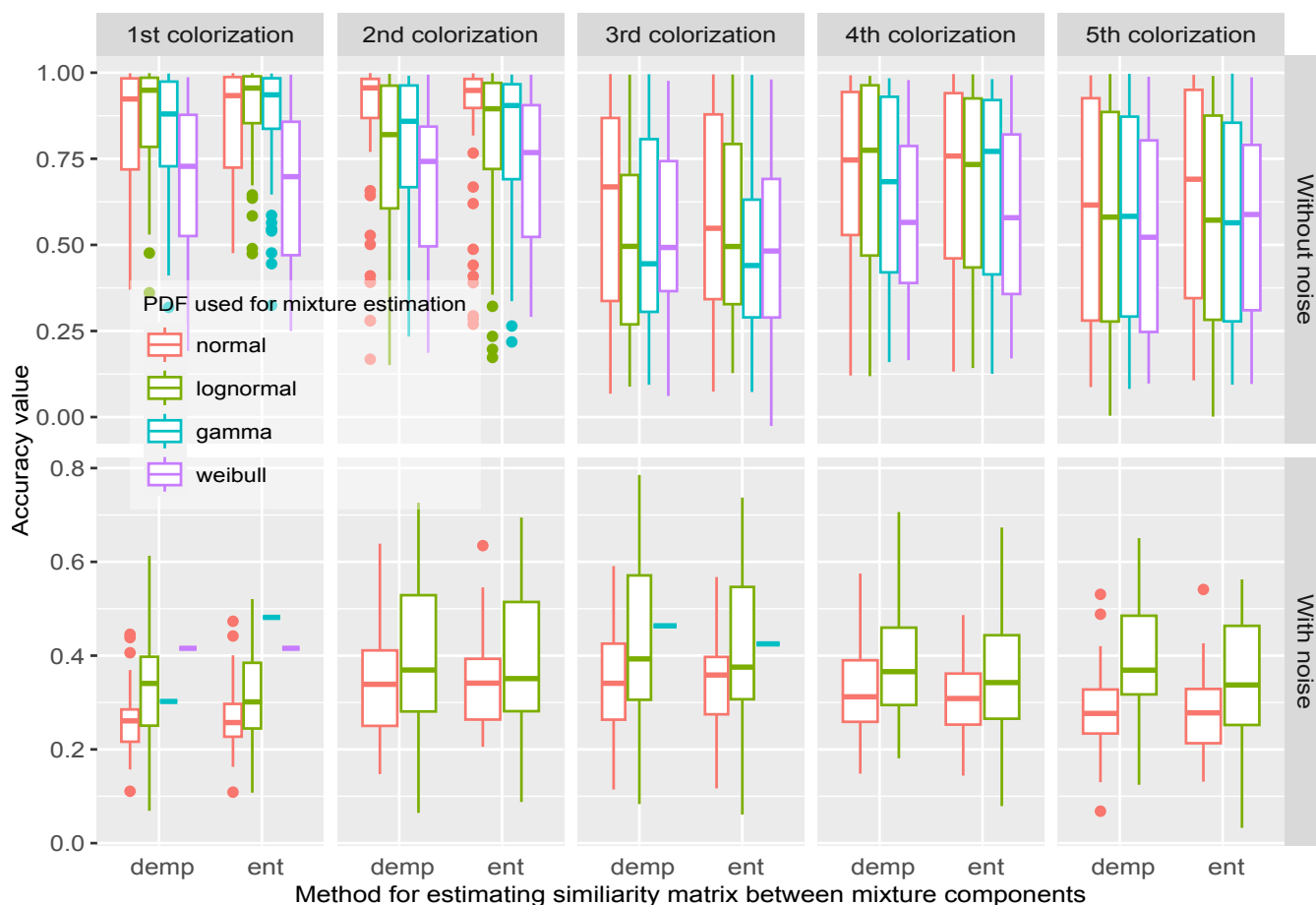


Figure 7. Boxplots of estimated accuracy values (ARI metric) using Markov random fields (MRFs) as an intermediate step. Results are grouped by the PDF used to estimate the mixture model, the method used to estimate the similarity matrix (adjacency matrix), the absence/presence of noise on the image, and different colorization.

6.3. Impact of Weight Parameter in Constructing the Adjacency Matrix

In the following, we focus on the weighting parameter used to create the final similarity matrix for spectral clustering. The ARI metric is estimated for different weights and averaged over all images (different shapes and different colorizations). The results are shown in Figure 8, where the accuracy value (i.e., the average ARI value over multiple images) is plotted against the value of the weighting used to create the final similarity matrix for spectral clustering.

With noise-free images (the two upper plots), the weighting parameter does not appear to have any significant effect. The accuracy seems to decrease significantly only when we strongly favor spatial similarity over color similarity (i.e., higher weighting values mean that the final adjacency matrix used for spectral clustering is mainly created from the spatial adjacency matrix). For noisy images (the bottom two plots), it is clear that the accuracy values increase with increasing weight, which ultimately means that we obtain better results when we favor spatial coherence. This could also be related to our chosen accuracy metric (ARI), which also gives better values for more compact blobs.

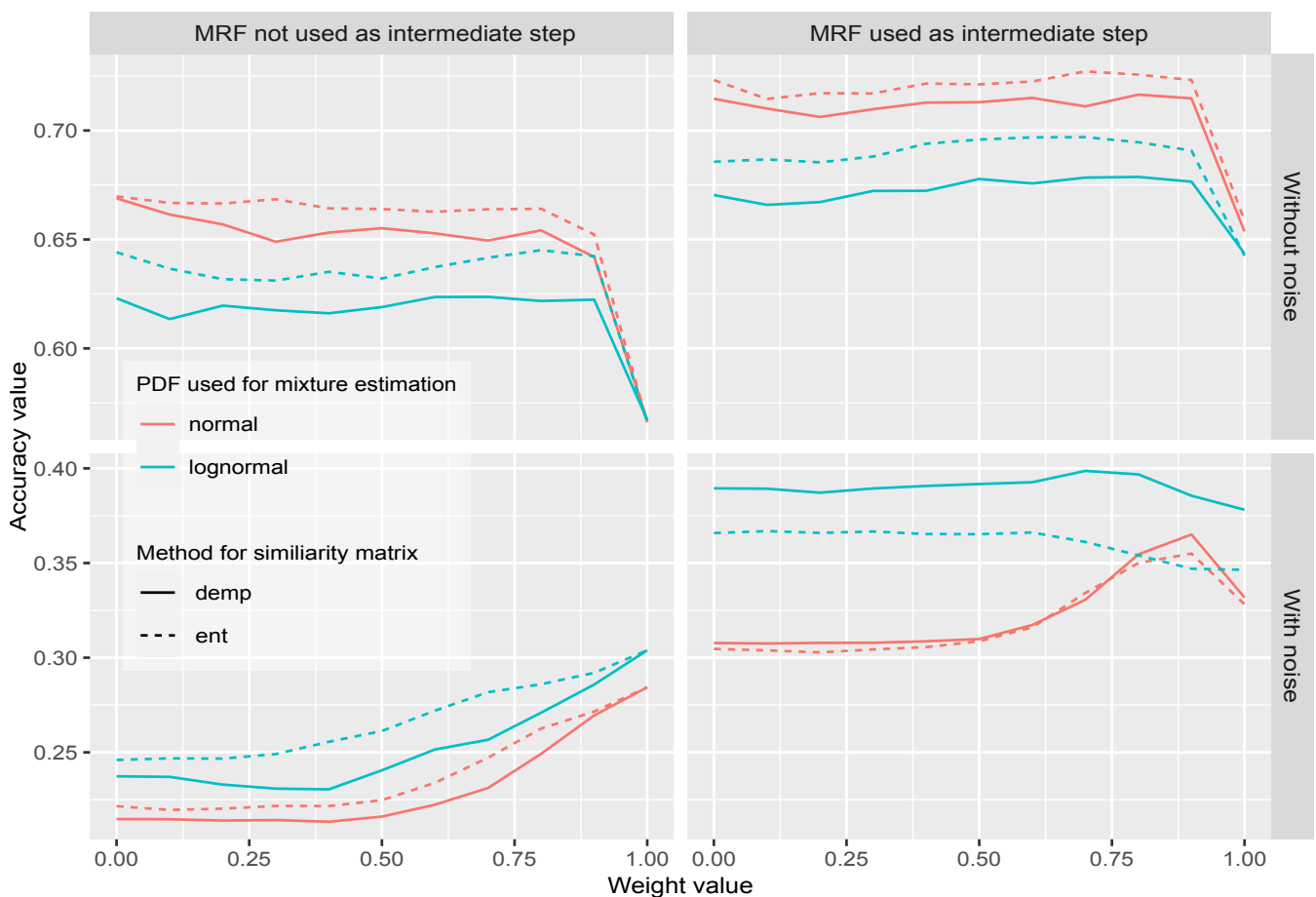


Figure 8. Impact of weight value α on estimated accuracy values (ARI metric).

6.4. Impact of the Smoothing Parameter and the Weight Parameter

The next reports deal with the weighting parameter α and the smoothing parameter σ . The smoothing value was used in Equation (8) to create the spatial adjacency matrix. The results are shown in Figure 9. It shows contours with averaged ARI values for all images (colorization and noise) for the two most interesting PDF components and both methods for creating the color adjacency matrix. We have chosen 0.3–3 for the range of smoothing values and 0–1 for the weights, as before. Since, in all our previous reports, the use of MRF as an intermediate step led to better results, we only report the results with MRF as an intermediate step.

Obviously, the influence of the smoothing and the weighting value is not very large. The largest difference was about 0.11 between the estimated average ARI value. The lowest value was 0.44 with a weighting value of 1, a smoothing value of 0.3, a demp similarity matrix, and a normal PDF. On the other hand, the highest value of 0.55 was obtained with a weighting value of 0.7 and a smoothing value of 0.3, a demp similarity matrix, and the lognormal PDF. The maximum value for the normal PDF was 0.545 for the ent method and a weighting value and sigma of 0.8. The minimum value for the lognormal PDF was 0.475 with ent with a weighting value of 1 and a sigma value of 0.3. It is clear that the effect of the “spurious” distribution can be mitigated by a different choice of weighting and sigma parameters and also by using a different method for estimating the color adjacency matrix (e.g., by using ent versus demp and vice versa). It is also clear that when using a larger smoothing value, the accuracy tends to be lower, but a wider range of weights can be used to obtain equivalent results. On the other hand, when using smaller values for the sigma parameter, we tend to obtain higher accuracy values (i.e., ARI) but need to be more critical in the choice of weight.

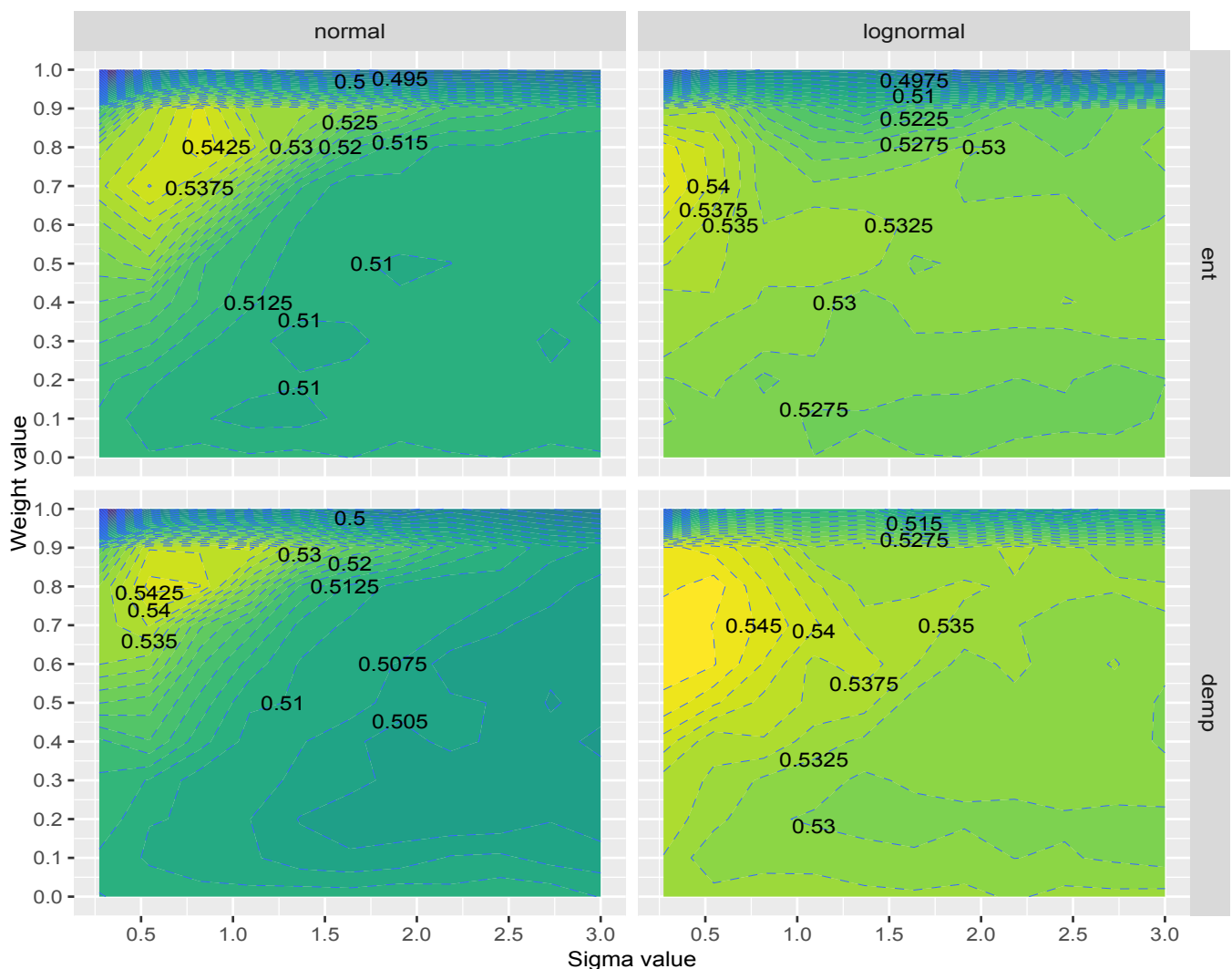


Figure 9. Impact of different smoothing values σ and weighting value α on estimated accuracy values (ARI metric). Results are grouped according to the PDF used for mixture estimation and the method used to create the color adjacency matrix.

6.5. Accuracy Comparisons with Other Unsupervised Clustering Algorithms for Image Segmentation

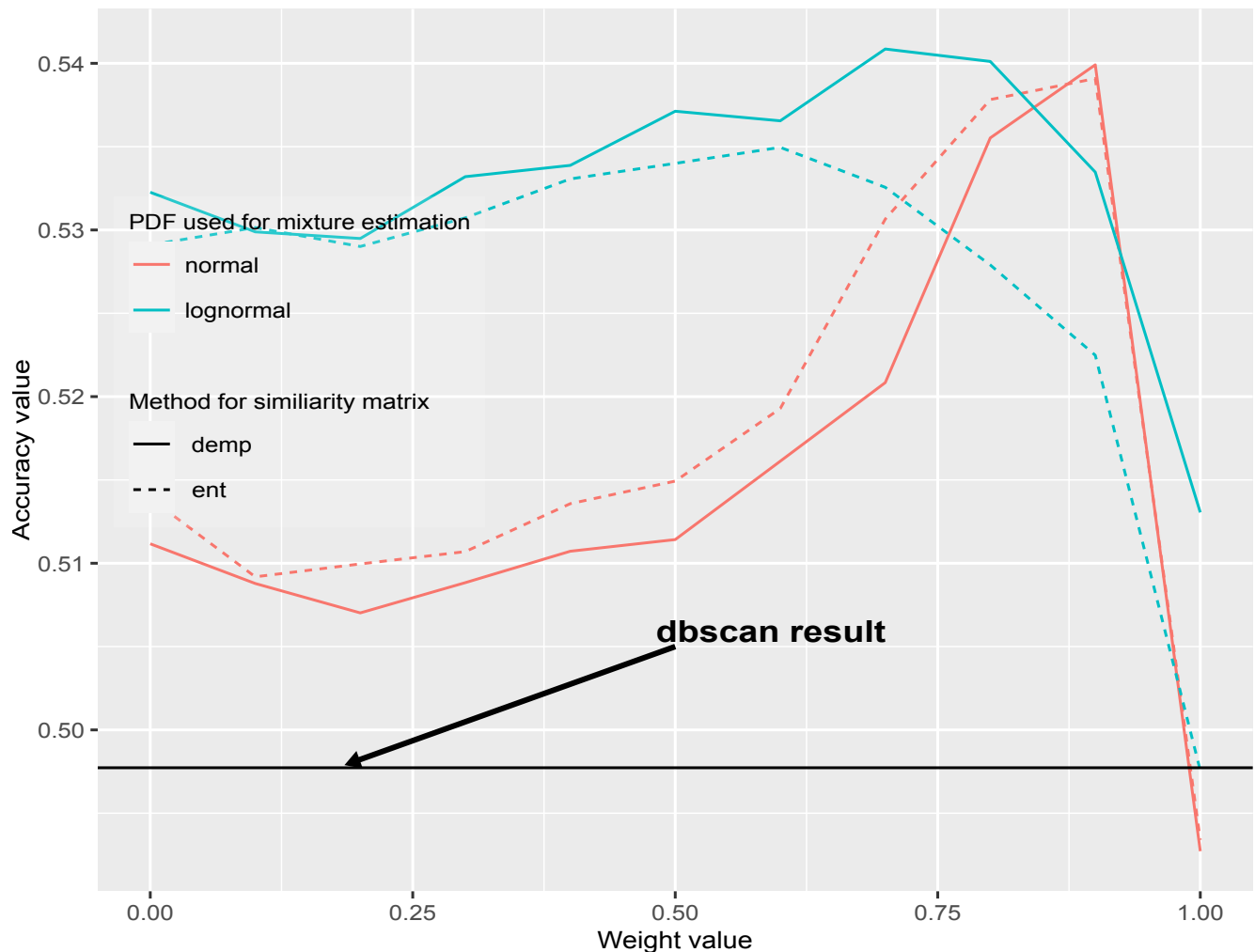
In addition, we have made some comparisons with other common clustering algorithms for unsupervised image segmentation. The algorithms used, k-means [52], mean-shift [53], dbscan [54], and optics [55], are all available and implemented in Python in the famous scikit-learn library [56]. To obtain the results in a reasonable time, we followed the strategy used by other authors [13]. First, we construct superpixel solutions using the SLIC algorithm [57], which is included in the scikit-image Python library [58]. The positions and intensities of the superpixels are then used for the final clustering. In this way, we shrink the input image from its original size (240×240) to the number of superpixels, which we set to 1000. As in our proposal, we use both spatial and color values.

The estimated ARI values are averaged over the entire dataset (different colorizations and noise) and the obtained values are shown in Table 1. For comparison with the results of our proposals, we show in Figure 10 the best ARI value obtained with the clustering algorithm dbscan as a horizontal line and the averaged ARI values for different weighting parameters and simply keeping the smoothing parameter $\sigma = 1$. It can be clearly seen that the accuracy achieved with our proposals is better for almost the entire range of weighting values.

Table 1. Results of the different clustering algorithms mainly used for unsupervised image segmentation.

	Mean-Shift	k-Means	Optics	Dbscan
Accuracy value ¹	0.06	0.15	0.37	0.49

¹ ARI is used as accuracy value.

**Figure 10.** Comparisons with dbscan algorithm. For accuracy metric, ARI is used.

6.6. Accuracy Comparisons with Convolutional Neural Networks for Supervised Image Segmentation

Finally, we have added some results obtained with state-of-the-art deep learning neural networks. Since we had no luck finding a suitable convolutional neural network that does not need to be trained and can be used immediately for unsupervised image segmentation, we used some of the best supervised convolutional neural networks for image segmentation. We chose the Attention U-Net convolutional neural network explained in [59] and implemented in Python using the PyTorch module [60]. Since the Attention U-Net was to be trained with labelled images, we selected our fifth colorization for training, as it would contain the most differences and features to be learned. Since training eleven classes with only 50 pictures seemed somewhat unrealistic, we only trained for two classes. All shapes were grouped into one class and the second class was the background. The resulting binary segmentation was then further segmented by applying connected component algorithms that split all shapes that were not connected into a separate segment. This was appropriate in this case, as our original simulated images were such that there was no overlap between the shapes.

First, we trained with images that did not contain noise and achieved a mean ARI (i.e., mean accuracy) of 0.84. In the second run, we trained with images that contained noise and achieved a mean ARI (i.e., mean accuracy) of 0.94. Both results were significantly better than the best result obtained with our proposal. However, this was to be expected, as the network learned almost 35 million parameters to discriminate between the image features, as opposed to the 100–200 parameters (mainly component parameters from the estimated mixture) used in our proposal. It is also quite difficult to properly evaluate the impact of the training dataset, as it is not always available and sometimes very complicated to obtain. Since unsupervised methods usually perform worse than supervised methods, we consider our proposal to be competitive in this comparison as well. This becomes clear when looking at the results in Figure 7, where some of the results are very clearly above the 0.9 mark.

7. Conclusions and Future Considerations

We conclude this study with the following remarks. In this work, we have proposed a strategy for using spectral clustering with mixture models for the purpose of unsupervised image segmentation. First, we encode the single image into a mixture model and use this model to estimate adjacency matrices that capture the color and spatial similarity in the image. Then we use spectral clustering to propose the final segmentation. As an intermediate step, we propose Markov random fields that can help with scattered segments due to noise.

Our proposal provided promising results that outperform the current state of the art in unsupervised image segmentation. Unfortunately, the current state of the art in supervised image segmentation, such as the Attention U-Net convolutional neural network, is still very far ahead. Although our proposed method is still inferior to the convolutional neural network in terms of accuracy, it has many advantages. It can be easily applied to a variety of problems, requires no training and consequently no training data, requires fewer parameters, which directly translates into lower computational costs, and so on. Ultimately, the added value of such comparisons is highly questionable, as the final intended use of the two methods is very different. While supervised image segmentation with convolutional neural networks focuses more on inference on many images that are similar to those used for training, unsupervised image segmentation focuses more on pattern recognition and further image analysis.

Finally, some interesting points have emerged in the course of this study. The first again concerns the estimation of the mixture model. Here we have not constrained any of the estimated parameters of the mixture model. However, it is quite common to impose restrictions on some parameters because different patterns could be found in the data [61]. We found that the “width” of the estimated component was inappropriate, resulting in too much scatter in the component’s color features, which could not be corrected at later stages. In addition, we again used only color values to estimate the mixture model. Although we consider this model to be quite robust for many examples, it still lacks some form of spatial coding to be even more robust, for example, the inclusion of local spatial information through the gradient intensity mixture model [62]. However, this requires further in-depth research. Second, we found that the labelling moments approach clearly benefited the segmentation procedure. However, it seemed to be limited in its ability to capture all the nuances of the shapes generated by the initial segmentation with the mixture model. It seems that more moments should be used that can also encode the orientation and some other spatial information about the segments. In this context, it might also be beneficial to include some kind of convolutional neural network for more precise shape matching and spatial adjacency matrix estimation, but this needs further research. The third point, which is also partly related to the second point, is the use of only two adjacency matrices. In this work, we used only two adjacency matrices to create the final adjacency matrix for spectral clustering. However, it is easily extendable to more adjacency matrices, which in general could provide more information and thus better final segmentation. Finally, we would like

to point out that our proposal currently relies on the manual selection of values for the smoothing and weighting parameters (σ and α). Although we have reported extensively on the effects of these parameters, it would be good if an estimation procedure for their determination could be proposed.

Author Contributions: All authors contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge financial support from the Slovenian Research Agency (research core funding No. P2-0182 entitled Development Evaluation).

Data Availability Statement: Data can be made available on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Borovinšek, M.; Taherishargh, M.; Vesenjaj, M.; Ren, Z.; Fiedler, T. Geometrical characterization of perlite-metal syntactic foam. *Mater. Charact.* **2016**, *119*, 209–215. [CrossRef]
2. He, Y.; Zhang, W.; Li, Y.F.; Wang, Y.L.; Wang, Y.; Wang, S.L. An approach for surface roughness measurement of helical gears based on image segmentation of region of interest. *Measurement* **2021**, *183*, 109905. [CrossRef]
3. Li, L.; An, Q. An in-depth study of tool wear monitoring technique based on image segmentation and texture analysis. *Measurement* **2016**, *79*, 44–52. [CrossRef]
4. Saxena, N.; Hofmann, R.; Alpak, F.O.; Dietderich, J.; Hunter, S.; Day-Stirrat, R.J. Effect of image segmentation & voxel size on micro-CT computed effective transport & elastic properties. *Mar. Pet. Geol.* **2017**, *86*, 972–990.
5. Dong, L.; Chen, W.; Yang, S.; Yu, H. Automated detection of gear tooth flank surface integrity: A cascade detection approach using machine vision. *Measurement* **2023**, *220*, 113375. [CrossRef]
6. Tang, C.; Liang, B.; Zhang, W. Detect and visualize non-uniform yarn orientations on preformed CFRP parts using automatic scanning and image processing. *J. Manuf. Process.* **2023**, *102*, 1043–1058. [CrossRef]
7. Zeng, S.; Huang, R.; Kang, Z.; Sang, N. Image segmentation using spectral clustering of Gaussian mixture models. *Neurocomputing* **2014**, *144*, 346–356. [CrossRef]
8. Panić, B.; Nagode, M.; Klemenc, J.; Oman, S. On Methods for Merging Mixture Model Components Suitable for Unsupervised Image Segmentation Tasks. *Mathematics* **2022**, *10*, 4301. [CrossRef]
9. Li, S.Z. *Markov Random Field Modeling in Image Analysis*; Springer Science & Business Media: London, UK, 2009.
10. Zhang, J. The mean field theory in EM procedures for Markov random fields. *IEEE Trans. Signal Process.* **1992**, *40*, 2570–2583. [CrossRef]
11. Wei, T.; Wang, X.; Li, X.; Zhu, S. Fuzzy subspace clustering noisy image segmentation algorithm with adaptive local variance & non-local information and mean membership linking. *Eng. Appl. Artif. Intell.* **2022**, *110*, 104672.
12. Panić, B.; Borovinšek, M.; Vesenjaj, M.; Oman, S.; Nagode, M. A Guide to Unsupervised Image Segmentation of Mct-Scanned Cellular Metals with Mixture Modelling and Markov Random Fields. Available online: <https://ssrn.com/abstract=4469707> (accessed on 27 September 2023).
13. Aksaç, A.; Özyer, T.; Alhajj, R. CutESC: Cutting edge spatial clustering technique based on proximity graphs. *Pattern Recognit.* **2019**, *96*, 106948. [CrossRef]
14. Stosic, D.; Stosic, D.; Ludermit, T.B.; Ren, T.I. Natural image segmentation with non-extensive mixture models. *J. Vis. Commun. Image Represent.* **2019**, *63*, 102598. [CrossRef]
15. Katunin, A.; Lis, K.; Jozsko, K.; Zak, P.; Dragan, K. Quantification of hidden corrosion in aircraft structures using enhanced D-Sight NDT technique. *Measurement* **2023**, *216*, 112977. [CrossRef]
16. Guo, Y.; Şengür, A.; Akbulut, Y.; Shipley, A. An effective color image segmentation approach using neutrosophic adaptive mean shift clustering. *Measurement* **2018**, *119*, 28–40. [CrossRef]
17. Zhang, T.f.; Li, Z.; Yuan, Q.; Wang, Y.n. A spatial distance-based spatial clustering algorithm for sparse image data. *Alex. Eng. J.* **2022**, *61*, 12609–12622. [CrossRef]
18. Liu, H.; Zhao, F.; Jiao, L. Fuzzy spectral clustering with robust spatial information for image segmentation. *Appl. Soft Comput.* **2012**, *12*, 3636–3647. [CrossRef]
19. Angulakshmi, M.; Priya, G.G.L. Brain tumour segmentation from MRI using superpixels based spectral clustering. *J. King Saud Univ.-Comput. Inf. Sci.* **2020**, *32*, 1182–1193.
20. Vacher, J.; Launay, C.; Coen-Cagli, R. Flexibly regularized mixture models and application to image segmentation. *Neural Netw.* **2022**, *149*, 107–123. [CrossRef]
21. Cheng, N.; Cao, C.; Yang, J.; Zhang, Z.; Chen, Y. A spatially constrained skew Student's mixture model for brain MR image segmentation and bias field correction. *Pattern Recognit.* **2022**, *128*, 108658. [CrossRef]
22. Ng, A.; Jordan, M.; Weiss, Y. On spectral clustering: Analysis and an algorithm. *Adv. Neural Inf. Process. Syst.* **2001**, *14*, 1–8.

23. Zong, B.; Song, Q.; Min, M.R.; Cheng, W.; Lumezanu, C.; Cho, D.; Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
24. Nguyen, D.M.; Vu, H.T.; Ung, H.Q.; Nguyen, B.T. 3D-brain segmentation using deep neural network and Gaussian mixture model. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 815–824.
25. Harsha, S.S.; Anne, K. Gaussian mixture model and deep neural network based vehicle detection and classification. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 17–25.
26. Lina, Y.C.; Chenb, P.C.; Wangc, C.H. Using deep learning and gaussian mixture models for road scene segmentation. *Int. J. Eng. Sci. Innov. Technol.* **2017**, *6*, 27–36.
27. Panić, B.; Klemenc, J.; Nagode, M. Improved initialization of the EM algorithm for mixture model parameter estimation. *Mathematics* **2020**, *8*, 373. [\[CrossRef\]](#)
28. Sun, H.; Yang, X.; Gao, H. A spatially constrained shifted asymmetric Laplace mixture model for the grayscale image segmentation. *Neurocomputing* **2019**, *331*, 50–57. [\[CrossRef\]](#)
29. Katunin, A.; Nagode, M.; Oman, S.; Cholewa, A.; Dragan, K. Monitoring of hidden corrosion growth in aircraft structures based on D-Sight inspections and image processing. *Sensors* **2022**, *22*, 7616. [\[CrossRef\]](#)
30. Shi, X.; Li, Y.; Zhao, Q. Flexible hierarchical Gaussian mixture model for high-resolution remote sensing image segmentation. *Remote Sens.* **2020**, *12*, 1219. [\[CrossRef\]](#)
31. Bäcklin, C.L.; Andersson, C.; Gustafsson, M.G. Self-tuning density estimation based on Bayesian averaging of adaptive kernel density estimations yields state-of-the-art performance. *Pattern Recognit.* **2018**, *78*, 133–143. [\[CrossRef\]](#)
32. Hennig, C. Methods for merging Gaussian mixture components. *Adv. Data Anal. Classif.* **2010**, *4*, 3–34. [\[CrossRef\]](#)
33. Baudry, J.P.; Raftery, A.E.; Celeux, G.; Lo, K.; Gottardo, R. Combining Mixture Components for Clustering. *J. Comput. Graph. Stat.* **2010**, *19*, 332–353. [\[CrossRef\]](#)
34. Schwarz, G. Estimating the dimension of a model. *Ann. Stat.* **1978**, *6*, 461–464. [\[CrossRef\]](#)
35. Shi, X.; Wang, Y.; Li, Y.; Dou, S. Remote Sensing Image Segmentation Based on Hierarchical Student’s t Mixture Model and Spatial Constrains with Adaptive Smoothing. *Remote Sens.* **2023**, *15*, 828. [\[CrossRef\]](#)
36. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905.
37. Bateson, M.; Lombaert, H.; Ben Ayed, I. Test-time adaptation with shape moments for image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Singapore, 18–22 September 2022; pp. 736–745.
38. da Silva, R.D.C.; Jenkyn, T.R.; Carranza, V.A. Enhanced pre-processing for deep learning in MRI whole brain segmentation using orthogonal moments. *Brain Multiphys.* **2022**, *3*, 100049. [\[CrossRef\]](#)
39. Abdulhussain, S.H.; Mahmmod, B.M.; Flusser, J.; AL-Utaibi, K.A.; Sait, S.M. Fast overlapping block processing algorithm for feature extraction. *Symmetry* **2022**, *14*, 715. [\[CrossRef\]](#)
40. Trombini, M.; Solarna, D.; Moser, G.; Dellepiane, S. A goal-driven unsupervised image segmentation method combining graph-based processing and Markov random fields. *Pattern Recognit.* **2023**, *134*, 109082. [\[CrossRef\]](#)
41. McLachlan, G.; Peel, D. *Finite Mixture Models*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2000.
42. Panić, B.; Klemenc, J.; Nagode, M. Optimizing the estimation of a histogram-bin width—Application to the multivariate mixture-model estimation. *Mathematics* **2020**, *8*, 1090. [\[CrossRef\]](#)
43. Tung, F.; Wong, A.; Clausi, D.A. Enabling scalable spectral clustering for image segmentation. *Pattern Recognit.* **2010**, *43*, 4069–4076. [\[CrossRef\]](#)
44. Hu, M.K. Visual pattern recognition by moment invariants. *IRE Trans. Inf. Theory* **1962**, *8*, 179–187.
45. Kyoya, S.; Yamanishi, K. Summarizing Finite Mixture Model with Overlapping Quantification. *Entropy* **2021**, *23*, 1503. [\[CrossRef\]](#)
46. Freguglia, V.; Garcia, N.L. Inference Tools for Markov Random Fields on Lattices: The R Package mrf2d. *J. Stat. Softw.* **2022**, *101*, 1–36. [\[CrossRef\]](#)
47. Besag, J. On the statistical analysis of dirty pictures. *J. R. Stat. Soc. Ser. B (Methodol.)* **1986**, *48*, 259–279. [\[CrossRef\]](#)
48. Konovalenko, I.; Maruschak, P.; Brezinová, J.; Viňáš, J.; Brezina, J. Steel surface defect classification using deep residual neural network. *Metals* **2020**, *10*, 846. [\[CrossRef\]](#)
49. Nagode, M.; Panić, B.; Klemenc, J.; Oman, S. Fault detection and classification with the rebmix R package. *Comput. Ind. Eng.* **2023**, *185*, 109628. [\[CrossRef\]](#)
50. Lam, S.K.; Pitrou, A.; Seibert, S. Numba: A llvm-based python jit compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, Austin, TX, USA, 15 November 2015; pp. 1–6.
51. Hubert, L.; Arabie, P. Comparing partitions. *J. Classif.* **1985**, *2*, 193–218. [\[CrossRef\]](#)
52. Franti, P.; Sieranoja, S. How much can k-means be improved by using better initialization and repeats? *Pattern Recognit.* **2019**, *93*, 95–112. [\[CrossRef\]](#)
53. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [\[CrossRef\]](#)

54. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
55. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [[CrossRef](#)]
56. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
57. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [[CrossRef](#)]
58. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [[CrossRef](#)]
59. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999.
60. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–12.
61. Celeux, G.; Govaert, G. Gaussian parsimonious clustering models. *Pattern Recognit.* **1995**, *28*, 781–793. [[CrossRef](#)]
62. Brenne, E.O.; Dahl, V.A.; Jørgensen, P.S. A physical model for microstructural characterization and segmentation of 3D tomography data. *Mater. Charact.* **2021**, *171*, 110796. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.